J 6

```
MMM         MMM  TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCCC  PPPPPPPPPPPP
MMM         MMM  TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCCC  PPPPPPPPPPPP
MMM         MMM  TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCCC  PPPPPPPPPPPP
MMMMMM   MMMMMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMMMMM   MMMMMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMMMMM   MMMMMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM        TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM        TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM        TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
MMM         MMM        TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
```

```
NN       NN  XX       XX  TTTTTTTTTT  VV       VV    000000    LL
NN       NN  XX       XX  TTTTTTTTTT  VV       VV    000000    LL
NN       NN  XX       XX      TT      VV       VV  00      00  LL
NN       NN  XX       XX      TT      VV       VV  00      00  LL
NNNN     NN    XX   XX        TT      VV       VV  00      00  LL
NNNN     NN    XX   XX        TT      VV       VV  00      00  LL
NN  NN   NN      XX            TT      VV       VV  00      00  LL
NN  NN   NN      XX            TT      VV       VV  00      00  LL
NN    NNNN     XX   XX        TT      VV       VV  00      00  LL
NN    NNNN     XX   XX        TT      VV       VV  00      00  LL
NN       NN  XX       XX      TT        VV   VV    00      00  LL              ....
NN       NN  XX       XX      TT        VV   VV    00      00  LL              ....
NN       NN  XX       XX      TT          VV        000000    LLLLLLLLLL      ....
NN       NN  XX       XX      TT          VV        000000    LLLLLLLLLL      ....

LL             IIIIII      SSSSSSSS
LL             IIIIII      SSSSSSSS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II      SS
LL               II        SSSSSS
LL               II        SSSSSS
LL               II              SS
LL               II              SS
LL               II              SS
LL               II              SS
LLLLLLLLLL     IIIIII    SSSSSSSS
LLLLLLLLLL     IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE NXTVOL (LANGUAGE (BLISS32) ,
;     2          0002  0                  IDENT = 'V04-000'
;     3          0003  0                  ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !*****************************************************************************
;     7          0007  1 !*                                                                           *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                      *
;    11          0011  1 !*                                                                           *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    16          0016  1 !*  OTHEP PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    17          0017  1 !*  TRANSFERRED.                                                              *
;    18          0018  1 !*                                                                           *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    21          0021  1 !*  CORPORATION.                                                              *
;    22          0022  1 !*                                                                           *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*****************************************************************************
;    28          0028  1
;    29          0029  1 !++
;    30          0030  1 !
;    31          0031  1 ! FACILITY:  MTAACP
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !       This module gets the next volume for read and write
;    35          0035  1 !
;    36          0036  1 !
;    37          0037  1 ! ENVIRONMENT:
;    38          0038  1 !
;    39          0039  1 !       VMS operating system, including privileged system services
;    40          0040  1 !       and internal exec routines.
;    41          0041  1 !
;    42          0042  1 !--
;    43          0043  1
;    44          0044  1 !
;    45          0045  1 !
;    46          0046  1 ! AUTHOR:  D. H. GILLESPIE,      CREATION DATE:  20-AUG-1977
;    47          0047  1 !
;    48          0048  1 ! MODIFIED BY:
;    49          0049  1 !
;    50          0050  1 !       V03-010 HH0041          Hai Huang                   24-Jul-1984
;    51          0051  1 !               Remove REQUIRE 'LIBD$:[VMSLIB.OBJ]MOUNTMSG.B32'.
;    52          0052  1 !
;    53          0053  1 !       V03-009 MMD0287         Meg Dumont,     10-Apr-1984  14:14
;    54          0054  1 !               Fix to the $MTACCESS return code where the ACCESS field
;    55          0055  1 !               could have gotten set to normal processing before
;    56          0056  1 !               all the errors were checked.
;    57          0057  1 !
```

```
;    58          0058  1 !       V03-008 LMP0221         L. Mark Pilant,         28-Mar-1984  14:50
;    59          0059  1 !               Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
;    60          0060  1 !               ORB$W_PROT.
;    61          0061  1 !
;    62          0062  1 !       V03-007 MMD0273         Meg Dumont,     23-Mar-1984  9:42
;    63          0063  1 !               Change the processing of the accessibility character fields
;    64          0064  1 !               in the VOL1 and or HDR1 label to call the installation
;    65          0065  1 !               specific accessibility routine. The return from this
;    66          0066  1 !               routine determines the users access to the volume and/or file.
;    67          0067  1 !
;    68          0068  1 !       V03-006 MMD0177         Meg Dumont,     26-May-1983  15:13
;    69          0069  1 !               Change VOL1 to indicate ANSI level 4 when writing SYSTEM CODE
;    70          0070  1 !               in VOL1 LABEL
;    71          0071  1 !
;    72          0072  1 !       V03-005 MMD0159         Meg Dumont,     26-Apr-1983  9:30
;    73          0073  1 !               Change reference to 240 the symbol SCRATCH_OFFSET.
;    74          0074  1 !
;    75          0075  1 !       V03-004 MMD0135         Meg Dumont,     12-Apr-1983  17:29
;    76          0076  1 !               Added support for writng and interrupting the VOL1
;    77          0077  1 !               OWNER IDENTIFIER field, so that it is no longer
;    78          0078  1 !               treated as a VMS field, strictly.
;    79          0079  1 !
;    80          0080  1 !
;    81          0081  1 !       V03-003 MMD0121         Meg Dumont,     29-Mar-1983  0:45
;    82          0082  1 !               Added support for the VOL2 label inside the MTAACP
;    83          0083  1 !
;    84          0084  1 !       V03-002 MMD0104         Meg Dumont,     17-Feb-1983  13:19
;    85          0085  1 !               Use GET_DEV_NAME for tape units name. Added code for AVR and AVL
;    86          0086  1 !
;    87          0087  1 !       V02-015 DMW00060        David Michael Walp      7-Dec-1981
;    88          0088  1 !               Rename TRANSLATION_TABLE to ANSI_A_GOOD
;    89          0089  1 !
;    90          0090  1 !       V02-014 DMW00037        David Michael Walp      17-Sep-1981
;    91          0091  1 !               Set MVL entry used when GTNEXT_VOL_READ places the label
;    92          0092  1 !               in the MVL
;    93          0093  1 !
;    94          0094  1 !       V02-013 DMW00031        David Michael Walp      18-Aug-1981
;    95          0095  1 !               Volume Access project
;    96          0096  1 !
;    97          0097  1 !       V02-012 DMW00018        David Michael Walp      20-May-1981
;    98          0098  1 !               Checks for File-Set-Id changed to look at the MVL rather
;    99          0099  1 !               then VCB ( 1st mounted volume label ).
;   100          0100  1 !
;   101          0101  1 !       V02-011 REFORMAT        Maria del C. Nasr       30-Jun-1980
;   102          0102  1 !
;   103          0103  1 !       A0010   MCN0003         Maria del C. Nasr       15-Oct-1979   9:26
;   104          0104  1 !               Add HDR3 processing
;   105          0105  1 !
;   106          0106  1 !       A0009   ACG0047         Andrew C. Goldstein,    9-Aug-1979  14:17
;   107          0107  1 !               Protection check interface changes
;   108          0108  1 !
;   109          0109  1 !**
;   110          0110  1
;   111          0111  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   112          0112  1 REQUIRE 'SRC$:MTADEF.B32';
;   113          0496  1
;   114          0497  1 LINKAGE
```

```
;  115     0498  1     CHECK_PROT          = JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2)
;  116     0499  1                             : NOTUSED (3, 4, 5, 6, 7, 8, 9, 10, 11),
;  117     0500  1     L$CHECK_HDR         = JSB : GLOBAL (SCRATCH = 9, CURRENT_VCB = 11)
;  118     0501  1                             NOTUSED (7, 8, 10);
;  119     0502  1
;  120     0503  1 FORWARD ROUTINE
;  121     0504  1     CHECK_HDR           : L$CHECK_HDR,        ! check that HDR can be overwritten
;  122     0505  1     GTNEXT_VOL_READ     : NOVALUE L$GTNEXT_VOL_RE,  ! get next volume for read
;  123     0506  1     GTNEXT_VOL_WRIT     : NOVALUE L$GTNEXT_VOL_WR,  ! get next volume for write
;  124     0507  1     INC_VOL_SECTION     : COMMON_CALL NOVALUE,      ! incr rel vol and sect #
;  125     0508  1     RESET_UNIT          : COMMON_CALL NOVALUE,
;  126     0509  1     UPDATE_MVL_LBL      : COMMON_CALL NOVALUE;      ! update label in MVL entry
;  127     0510  1
;  128     0511  1 EXTERNAL
;  129     0512  1     CURRENT_UCB         : REF BBLOCK,         ! addr current unit control block
;  130     0513  1     HDR1                : REF BBLOCK,         ! addr of HDR1(EOF1) label
;  131     0514  1     IO_PACKET           : REF BBLOCK,         ! addr current I/O request packet
;  132     0515  1     SCA$GL_PCBVEC       : REF VECTOR ADDRESSING_MODE (ABSOLUTE),
;  133     0516  1     WORK_AREA;
;  134     0517  1
;  135     0518  1 EXTERNAL ROUTINE
;  136     0519  1     EXPIRED             : COMMON_CALL,              ! determine if file has expired
;  137     0520  1     FORMAT_VOLOWNER     : NOVALUE,                  ! format the owner field in VOL2
;  138     0521  1     GET_DEV_NAME        : COMMON_CALL NOVALUE,      ! given UCB addr get dev name
;  139     0522  1     GET_RECORD,                                     ! get record tape is currently reading
;  140     0523  1     ISSUE_IO            : L$ISSUE_IO,               ! issue I/O
;  141     0524  1     MOUNT_VOL           : COMMON_CALL,              ! mount relative vol
;  142     0525  1     PRINT_OPR_MSG       : L$PRINT_OPR_MSG,          ! print a system mess for oper
;  143     0526  1     READ_HDR            : COMMON_CALL,              ! read headers
;  144     0527  1     REWIND_AND_WAIT     : COMMON_CALL;
```

```
;  146     0528  1 GLOBAL ROUTINE GTNEXT_VOL_READ : NOVALUE L$GTNEXT_VOL_RE =
;  147     0529  1
;  148     0530  1 !++
;  149     0531  1 !
;  150     0532  1 ! FUNCTIONAL DESCRIPTION:
;  151     0533  1 !       This routine gets the next volume for read and checks that the file
;  152     0534  1 !       sequence number, file section number and volume set identifier
;  153     0535  1 !       are those sought
;  154     0536  1 !
;  155     0537  1 ! CALLING SEQUENCE:
;  156     0538  1 !       GTNEXT_VOL_READ()
;  157     0539  1 !
;  158     0540  1 ! INPUT PARAMETERS:
;  159     0541  1 !       NONE
;  160     0542  1 !
;  161     0543  1 ! IMPLICIT INPUTS:
;  162     0544  1 !       CURRENT_VCB - address of current volume control block
;  163     0545  1 !
;  164     0546  1 ! OUTPUT PARAMETERS:
;  165     0547  1 !       NONE
;  166     0548  1 !
;  167     0549  1 ! IMPLICIT OUTPUTS:
;  168     0550  1 !       next relative volume mounted
;  169     0551  1 !
;  170     0552  1 ! ROUTINE VALUE:
;  171     0553  1 !       NONE
;  172     0554  1 !
;  173     0555  1 ! SIDE EFFECTS:
;  174     0556  1 !       NONE
;  175     0557  1 !
;  176     0558  1 !--
;  177     0559  1
;  178     0560  2     BEGIN
;  179     0561  2
;  180     0562  2     EXTERNAL REGISTER
;  181     0563  2         COMMON_REG;
;  182     0564  2
;  183     0565  2     LOCAL
;  184     0566  2         CVT_DEVNAM      : VECTOR [MAX_DEVNAM_LENGTH,BYTE], ! Converted dev name
;  185     0567  2         CVT_DEVNAM_LENGTH : BYTE,                  ! and length of dev name
;  186     0568  2         VOL[BL  : BBLOCK [6],           ! current tape volume label
;  187     0569  2         FLAGS,
;  188     0570  2         FID,                            ! file identifier
;  189     0571  2         MVL_ENTRY : REF BBLOCK,         ! addr of current rel vol entry in MVL
;  190     0572  2         RVN,                            ! current relative volume number
;  191     0573  2         MVL     : REF BBLOCK;           ! magnetic tape volume list
;  192     0574  2
;  193     0575  2     FLAGS = $FIELDMASK(MOU$V_REWIND) OR $FIELDMASK(MOU$V_CHKIFSPC);
;  194     0576  2     KERNEL_CALL(INC_VOL_SECTION);       ! incr sequence # and relative vol #
;  195     0577  2     FID = .CURRENT_VCB[VCB$L_CUR_FID];  ! pickup current file id
;  196     0578  2     RVN = .CURRENT_VCB[VCB$B_CUR_RVN];  ! pickup cur relative volume #
;  197     0579  2
;  198     0580  2     WHILE 1
;  199     0581  2     DO
;  200     0582  3         BEGIN
;  201     0583  3
;  202     0584  3         LOCAL
```

```
;  203     0585  3          SCRATCH     : REF BBLOCK;
;  204     0586  3
;  205     0587  3      ! mount vol, rewind it, check the label if the operator specifies it
;  206     0588  3
;  207     0589  3      MVL_ENTRY = MOUNT_VOL(.RVN, .FLAGS);
;  208     0590  3
;  209     0591  3      SCRATCH = .HDR1 + SCRATCH_OFFSET;
;  210     0592  3      CH$MOVE(VLI$S_VOLLBL, SCRATCH[VLI$T_VOLLBL], VOLLBL);
;  211     0593  3
;  212     0594  3      IF NOT READ_HDR()
;  213     0595  3      THEN
;  214     0596  4          BEGIN
;  215     0597  4          ERR_EXIT(SS$_TAPEPOSLOST);
;  216     0598  3          END;
;  217     0599  3
;  218     0600  3      ! This next call will use the UCB address to get the device's name and
;  219     0601  3      ! will fill in the fields with that name and the length of the name.
;  220     0602  3
;  221     0603  3      GET_DEV_NAME(CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;  222     0604  3
;  223     0605  3      ! on read the next volume has the same volume set id and the fid of the
;  224     0606  3      ! next section for the current file
;  225     0607  3
;  226     0608  3      IF .FID NEQ .CURRENT_VCB[VCB$L_CUR_FID]
;  227     0609  3      THEN
;  228     0610  3          PRINT_OPR_MSG(MOUN$_NOTRELVOL, 0, .CURRENT_VCB[VCB$B_CUR_RVN],
;  229     0611  3                        .CVT_DEVNAM_LENGTH,CVT_DEVNAM)
;  230     0612  3
;  231     0613  3      ELSE
;  232     0614  4          BEGIN
;  233     0615  4
;  234     0616  4          ! pickup the addr of the MVL
;  235     0617  4
;  236     0618  4          MVL = .CURRENT_VCB[VCB$L_MVL];
;  237     0619  4          IF CH$NEQ(MVL$S_SET_ID, MVL[MVL$T_SET_ID],
;  238     0620  4                    HD1$S_FILESETID, HDR1[HD1$T_FILESETID], ' ')
;  239     0621  4            AND
;  240     0622  4
;  241     0623  4              ! not override set identifier with privs
;  242     0624  4
;  243     0625  5              NOT (      .CURRENT_VCB[VCB$V_OVRSETID]
;  244     0626  5                    AND .MVL_ENTRY [MVL$V_OVERIDE])
;  245     0627  5
;  246     0628  4          THEN
;  247     0629  4              PRINT_OPR_MSG(MOUN$_NOTVOLSET, 0,
;  248     0630  4                            .CVT_DEVNAM_LENGTH,CVT_DEVNAM,
;  249     0631  4                            6, MVL[MVL$T_SET_ID])
;  250     0632  4
;  251     0633  4          ELSE
;  252     0634  4              EXITLOOP;
;  253     0635  4
;  254     0636  3          END;
;  255     0637  3
;  256     0638  3      FLAGS = $FIELDMASK(MOU$V_REWIND) + $FIELDMASK(MOU$V_MOUNTERR);
;  257     0639  3      KERNEL_CALL(RESET_UNIT);
;  258     0640  2      END;                                    ! end of while loop
;  259     0641  2
```

```
;  260          0642  2     KERNEL_CALL(UPDATE_MVL_LBL, .MVL_ENTRY, VOLLBL);
;  261          0643  1     END;                                    ! end of routine


                                                        .TITLE  NXTVOL
                                                        .IDENT  \V04-000\

                                                        .EXTRN  CURRENT_UCB, HDR1
                                                        .EXTRN  IO_PACKET, SCH$GL_PCBVEC
                                                        .EXTRN  WORK_AREA, EXPIRED
                                                        .EXTRN  FORMAT_VOLOWNER
                                                        .EXTRN  GET_DEV_NAME, GET_RECORD
                                                        .EXTRN  ISSUE_IO, MOUNT_VOL
                                                        .EXTRN  PRINT_OPR_MSG, READ_HDR
                                                        .EXTRN  REWIND_AND_WAIT
                                                        .EXTRN  SYS$CMKRNL

                                                        .PSECT  $CODE$,NOWRT,2

                                07FC   8F   BB 00000 GTNEXT_VOL_READ::
                                                        PUSHR   #^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>        ; 0528
                          5E                1C   C2 00004         SUBL2   #28, SP                         ;
                          58                05   D0 00007         MOVL    #5, FLAGS                       ; 0575
                                            7E   D4 0000A         CLRL    -(SP)                           ; 0576
                                            5E   DD 0000C         PUSHL   SP                              ;
                                    0000V   CF   9F 0000E         PUSHAB  INC_VOL_SECTION                 ;
                    00000000G       9F           03   FB 00012    CALLS   #3, @#SYS$CMKRNL                ;
                              5A      24         AB   D0 00019    MOVL    36(CURRENT_VCB), FID            ; 0577
                              59      2F         AB   9A 0001D    MOVZBL  47(CURRENT_VCB), RVN            ; 0578
                                                 58   DD 00021 1$: PUSHL   FLAGS                         ; 0589
                                                 59   DD 00023    PUSHL   RVN                             ;
                                0000G   CF               02   FB 00025    CALLS   #2, MOUNT_VOL           ;
                                        57               50   D0 0002A    MOVL    R0, MVL_ENTRY           ;
                          50            0000G   CF 00000140   8F   C1 0002D    ADDL3   #320, HDR1, SCRATCH     ; 0591
                    04    AE              04    A0           06   28 00037    MOVC3   #6, 4(SCRATCH), VOLLBL  ; 0592
                                0000G   CF                   00   FB 0003D    CALLS   #0, READ_HDR            ; 0594
                                        04                   50   E8 00042    BLBS    R0, 2$                  ;
                                                      0224   8F   BF 00045    CHMU    #548                    ; 0597
                                                        0C   AE   9F 00049 2$:  PUSHAB  CVT_DEVNAM              ; 0603
                                                        04   AE   9F 0004C    PUSHAB  CVT_DEVNAM_LENGTH       ;
                                0000G   CF                   02   FB 0004F    CALLS   #2, GET_DEV_NAME        ;
                                  24    AB                   5A   D1 00054    CMPL    FID, 36(CURRENT_VCB)    ; 0608
                                                             1B   13 00058    BEQL    3$                      ;
                                                        0C   AE   9F 0005A    PUSHAB  CVT_DEVNAM              ; 0610
                                                 7E     04   AE   9A 0005D    MOVZBL  CVT_DEVNAM_LENGTH, -(SP) ; 0611
                                                 7E     2F   AB   9A 00061    MOVZBL  47(CURRENT_VCB), -(SP)  ; 0610
                                                             7E   D4 00065    CLRL    -(SP)                   ;
                                                00728114     8F   DD 00067    PUSHL   #7504148                ;
                                                      0000G  30   00006D       BSBW    PRINT_OPR_MSG           ;
                                                 5E                 14   C0 00070    ADDL2   #20, SP         ;
                                                                    35   11 00073    BRB     5$              ;
                                                 56     34         AB   D0 00075 3$:  MOVL    52(CURRENT_VCB), MVL    ; 0618
                                                 50     0000G      CF   D0 00079    MOVL    HDR1, R0        ; 0620
                    15    A0              0C    A6                 06   29 0007E    CMPC3   #6, 12(MVL), 21(R0)     ;
                                                                    39   13 00084    BEQL    6$              ;
                          05              2C    AB                 03   E1 00086    BBC     #3, 44(CURRENT_VCB), 4$ ; 0625
                          2F              07    A7                 02   E0 0008B    BBS     #2, 7(MVL_ENTRY), 6$    ; 0626
```

```
                      0C    A6  9F 00090 4$:     PUSHAB  12(MVL)                         ; 0631
                            06  DD 00093         PUSHL   #6                              ;
                      14    AE  9F 00095         PUSHAB  CVT_DEVNAM                      ; 0629
                   7E 0C    AE  9A 00098         MOVZBL  CVT_DEVNAM_LENGTH, -(SP)        ; 0631
                            7E  D4 0009C         CLRL    -(SP)                           ;
                0072810C    8F  DD 0009E         PUSHL   #7504140                        ;
                      0000G 30  000A4            BSBW    PRINT_OPR_MSG                   ;
                5E          18  C0 000A7         ADDL2   #24, SP                         ;
                58          09  D0 000AA 5$:     MOVL    #9, FLAGS                       ; 0638
                            7E  D4 000AD         CLRL    -(SP)                           ; 0639
                            5E  DD 000AF         PUSHL   SP                              ;
                      0000V CF  9F 000B1         PUSHAB  RESET_UNIT                      ;
          00000000G 9F          03  FB 000B5    CALLS   #3, @#SYS$CMKRNL                ;
                          FF62  31  000BC        BRW     1$                              ; 0580
                      04    AE  9F 000BF 6$:     PUSHAB  VOLLBL                          ; 0642
                            57  DD 000C2         PUSHL   MVL_ENTRY                       ;
                            02  DD 000C4         PUSHL   #2                              ;
                            5E  DD 000C6         PUSHL   SP                              ;
                      0000V CF  9F 000C8         PUSHAB  UPDATE_MVL_LBL                  ;
          00000000G 9F          05  FB 000CC    CALLS   #5, @#SYS$CMKRNL                ;
                5E          1C  C0 000D3         ADDL2   #28, SP                         ; 0643
                      07FC  8F  BA 000D6         POPR    #^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>;
                            05  000DA         RSB                                     ;

; Routine Size:  219 bytes,    Routine Base:  $CODE$ + 0000

;  262          0644  1
```

```
;  264     0645  1 GLOBAL ROUTINE GTNEXT_VOL_WRIT : NOVALUE L$GTNEXT_VOL_WR =
;  265     0646  1
;  266     0647  1 !++
;  267     0648  1 !
;  268     0649  1 ! FUNCTIONAL DESCRIPTION:
;  269     0650  1 !       This routine gets the next volume for write.  The volume
;  270     0651  1 !       is mounted, rewound and the label is verified. The VOL1
;  271     0652  1 !       label is rewritten to insure same density throughout volume set.
;  272     0653  1 !       The tape in initialized at the operator's request. The tape is
;  273     0654  1 !       also inited at the request of the user who mounted the tape.
;  274     0655  1 !       That is if the tape was mounted /INIT or /BLANK then every new
;  275     0656  1 !       reel in the volume set will be inited if the user has the proper
;  276     0657  1 !       access to the tape.
;  277     0658  1 !
;  278     0659  1 ! CALLING SEQUENCE:
;  279     0660  1 !       GTNEXT_VOL_WRIT()
;  280     0661  1 !
;  281     0662  1 ! INPUT PARAMETERS:
;  282     0663  1 !       NONE
;  283     0664  1 !
;  284     0665  1 ! IMPLICIT INPUTS:
;  285     0666  1 !       CURRENT_UCB - address of current unit control block
;  286     0667  1 !       CURRENT_VCB - address of current volume control block
;  287     0668  1 !       operator input
;  288     0669  1 !
;  289     0670  1 ! OUTPUT PARAMETERS:
;  290     0671  1 !       NONE
;  291     0672  1 !
;  292     0673  1 ! IMPLICIT OUTPUTS:
;  293     0674  1 !       relative volume number incremented
;  294     0675  1 !       section number increment
;  295     0676  1 !
;  296     0677  1 ! ROUTINE VALUE:
;  297     0678  1 !       NONE
;  298     0679  1 !
;  299     0680  1 ! SIDE EFFECTS:
;  300     0681  1 !       NONE
;  301     0682  1 !
;  302     0683  1 !--
;  303     0684  1
;  304     0685  2     BEGIN
;  305     0686  2
;  306     0687  2     LITERAL
;  307     0688  2         BLANK = 0,
;  308     0689  2         INIT = 1;
;  309     0690  2
;  310     0691  2     LOCAL
;  311     0692  2         CHAR     : VECTOR [4,BYTE],      ! Char to write in accessibility field
;  312     0693  2         CURRENT_RECORD,                  ! current record tape drive is reading
;  313     0694  2         CVT_DEVNAM       : VECTOR [MAX_DEVNAM_LENGTH,BYTE], ! Converted dev name
;  314     0695  2         CVT_DEVNAM_LENGTH : BYTE,                 ! and length of dev name
;  315     0696  2         ERROR_NO,
;  316     0697  2         FLAGS,
;  317     0698  2         ORB              : REF BBLOCK,           ! ORB address
;  318     0699  2         MVL              : REF BBLOCK,           ! MVL of current volume set
;  319     0700  2         MVL_ENTRY        : REF BBLOCK,           ! Entry of current volume
;  320     0701  2         SAVE_DEVCHAR     : VECTOR [2],
```

```
;  321          0702  2          OPR_FLAG        : BITVECTOR [2],
;  322          0703  2          ACCESS_CHAR     : BYTE,
;  323          0704  2          VOL_OWNER       : VECTOR [ VL1$S_OWNER_IDENT, BYTE],
;  324          0705  2          SCRATCH2        : BBLOCK [ANSI_LBLSZ],
;  325          0706  2          STATUS;
;  326          0707  2
;  327          0708  2      GLOBAL REGISTER
;  328          0709  2          SCRATCH = 9     : REF BBLOCK;
;  329          0710  2
;  330          0711  2      BIND
;  331          0712  2          MAIL = WORK_AREA        : BBLOCK [MSGSIZE],
;  332          0713  2          MAILSZ = MAIL + MSGSIZE,
;  333          0714  2          STARID = UPLIT ('DECFILE11A');
;  334          0715  2
;  335          0716  2      EXTERNAL REGISTER
;  336          0717  2          COMMON_REG;
;  337          0718  2
;  338          0719  2      KERNEL_CALL(INC_VOL_SECTION);
;  339          0720  2      SAVE_DEVCHAR[0] = .(CURRENT_UCB[UCB$B_DEVCLASS])<0, 32>;
;  340          0721  2      SAVE_DEVCHAR[1] = .CURRENT_UCB[UCB$L_DEVDEPEND];
;  341          0722  2      SCRATCH = .HDR1 + SCRATCH_OFFSET;
;  342          0723  2      FLAGS = $FIELDMASK(MOU$V_REWIND);
;  343          0724  2
;  344          0725  2      ! This next call will use the UCB address to get the device's name and
;  345          0726  2      ! will fill in the fields with that name and the length of the name.
;  346          0727  2
;  347          0728  2      GET_DEV_NAME(CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;  348          0729  2
;  349          0730  2      WHILE 1
;  350          0731  2      DO
;  351          0732  3          BEGIN
;  352          0733  3
;  353          0734  3          WHILE 1
;  354          0735  3          DO
;  355          0736  4              BEGIN
;  356          0737  4
;  357          0738  4              ! mount the volume, check if overwrite is possible
;  358          0739  4
;  359          0740  4              MVL_ENTRY = MOUNT_VOL(.CURRENT_VCB[VCB$B_CUR_RVN], .FLAGS);
;  360          0741  4              MVL = .CURRENT_VCB[VCB$L_MVL];
;  361          0742  4
;  362          0743  4              ! set operator flag for "/INIT" and "/BLANK". If the operator
;  363          0744  4              ! was required to intervene then these flags may be set.
;  364          0745  4
;  365          0746  5              OPR_FLAG [ BLANK ] = (.MAIL [ OPC$W_MS_STATUS ] EQL
;  366          0747  4                                      ( OPC$_BLANKTAPE AND %X'FFFF' ));
;  367          0748  5              OPR_FLAG [ INIT ]  = (.MAIL [ OPC$W_MS_STATUS ] EQL
;  368          0749  4                                      ( OPC$_INITAPE AND %X'FFFF' ));
;  369          0750  4
;  370          0751  4              ! do not check things on "/BLANK" or if the volume was mounted
;  371          0752  4              ! "/BLANK"
;  372          0753  4
;  373          0754  4              IF .OPR_FLAG[BLANK] OR .CURRENT_VCB[VCB$V_BLANK] THEN EXITLOOP;
;  374          0755  4
;  375          0756  4              ! see if we can overwrite the 1st file, save the VOL1 access
;  376          0757  4              ! character ( for defaulting ) before scratching the scratch area
;  377          0758  4
```

```
;  378     0759  4              ACCESS_CHAR = .SCRATCH [ VL1$B_VOLACCESS ];
;  379     0760  4              CH$MOVE (VL1$S_OWNER_IDENT, SCRATCH[VL1$T_OWNER_IDENT],VOL_OWNER);
;  380     0761  5              ERROR_NO    = CHECK_ADR(.MVL_ENTRY,(.OPR_FLAG[INIT]
;  381     0762  4                                      OR .CURRENT_VCB[VCB$V_INIT]));
;  382     0763  4
;  383     0764  4              ! check on the results
;  384     0765  4
;  385     0766  6              IF .ERROR_NO OR ((.OPR_FLAG[INIT] OR .CURRENT_VCB[VCB$V_INIT])
;  386     0767  5                           AND (.ERROR_NO EQL MOUN$_NOTANSIT)
;  387     0768  4                           THEN EXITLOOP;
;  388     0769  4
;  389     0770  4              ! the tape is not ANSI without /INIT or /BLANK
;  390     0771  4
;  391     0772  4              PRINT_OPR_MSG(.ERROR_NO, 0,
;  392     0773  4                            .CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;  393     0774  4
;  394     0775  4              ! force physical mount
;  395     0776  4
;  396     0777  4              FLAGS = $FIELDMASK(MOU$V_REWIND) + $FIELDMASK(MOU$V_MOUNTERR);
;  397     0778  4              KERNEL_CALL(RESET_UNIT);
;  398     0779  3              END;
;  399     0780  3
;  400     0781  3          ERROR_NO = MOUN$_IOERROR;
;  401     0782  3
;  402     0783  3          ! try to initialize
;  403     0784  3
;  404     0785  3          IF REWIND_AND_WAIT()
;  405     0786  3          THEN
;  406     0787  4              BEGIN
;  407     0788  4
;  408     0789  4              ! fill with spaces
;  409     0790  4
;  410     0791  4              CH$FILL(' ', ANSI_LBLSZ, .SCRATCH);
;  411     0792  4              CH$FILL(' ', ANSI_LBLSZ, SCRATCH2);
;  412     0793  4
;  413     0794  4              ! Set defaults
;  414     0795  4
;  415     0796  4              .SCRATCH = 'VOL1';
;  416     0797  4              SCRATCH[VL1$B_LBLSTDVER] = .MVL[MVL$B_STDVER] + '0';
;  417     0798  4              SCRATCH2 = 'VOL2';
;  418     0799  4              (SCRATCH2[VL2$T_VOLOWNER])<0,32> = 'D%C ';
;  419     0800  4
;  420     0801  4              ! get the volume label from the MVL
;  421     0802  4
;  422     0803  4              CH$COPY(MVL$S_VOLLBL, MVL_ENTRY[MVL$T_VOLLBL], ' ',
;  423     0804  4                      VL1$S_VOLLBL, SCRATCH[VL1$T_VOLLBL]);
;  424     0805  4
;  425     0806  4              ! If the operator supplied a label or if the MTAACP created
;  426     0807  4              ! the label, the ANSI volume owner from the MVL is stored in
;  427     0808  4              ! the label else the one currently on the tape will be used.
;  428     0809  4              ! The accessibility char to input to $MTACCESS is determined
;  429     0810  4              ! in a similar fashion, except it is not stored in the
;  430     0811  4              ! label until $MTACCESS has seen it.
;  431     0812  4
;  432     0813  4              IF (.MAILSZ NEQ 0) OR .OPR_FLAG [ INIT ] OR .OPR_FLAG [ BLANK ]
;  433     0814  4                                 OR .CURRENT_VCB [ VCB$V_INIT ]
;  434     0815  4                                 OR .CURRENT_VCB [ VCB$V_BLANK ]
```

```
;  435          0816  4          THEN
;  436          0817  5          BEGIN
;  437          0818  5              ACCESS_CHAR = .MVL[MVL$B_VOL_ACC];
;  438          0819  5              CH$MOVE(VL1$S_OWNER_IDENT, MVL[MVL$T_VOLOWNER],
;  439          0820  5                      SCRATCH[VL1$T_OWNER_IDENT]);
;  440          0821  5          END
;  441          0822  4          ELSE
;  442          0823  5          BEGIN
;  443          0824  5              CH$MOVE (VL1$S_OWNER_IDENT, VOL_OWNER,
;  444          0825  5                       SCRATCH [VL1$T_OWNER_IDENT]);
;  445          0826  4          END;
;  446          0827  4
;  447          0828  4          ! Call the accessibility system service to get the character to output.
;  448          0829  4          ! First keep the record that the UCB is reading. The accessibility
;  449          0830  4          ! routine can not move the tape from under us! Thus we will compare
;  450          0831  4          ! this to the field after the call and if the tape was moved we punt
;  451          0832  4          ! the operation.
;  452          0833  4
;  453          0834  4          ORB = .CURRENT_UCB[UCB$L_ORB];
;  454          0835  4          CURRENT_RECORD = KERNEL_CALL (GET_RECORD,.CURRENT_UCB);
;  455        P 0836  4          CHAR = $MTACCESS(LBLNAM = 0,
;  456        P 0837  4                           UIC = .ORB[ORB$L_OWNER],
;  457        P 0838  4                           STD_VERSION = .MVL[MVL$B_STDVER],
;  458        P 0839  4                           ACCESS_CHAR = .ACCESS_CHAR,
;  459        P 0840  4                           ACCESS_SPEC = MTA$K_CHARVALID,
;  460          0841  4                           TYPE = MTA$K_OUTVOLT);
;  461          0842  4
;  462          0843  4          STATUS = KERNEL_CALL( GET_RECORD, .CURRENT_UCB);
;  463          0844  4          IF .CURRENT_RECORD EQL .STATUS
;  464          0845  4           THEN
;  465          0846  5           BEGIN
;  466          0847  5               LOCAL   TMP_PROT        : WORD;         ! SOGW protection word
;  467          0848  5
;  468          0849  5               ! Set the access char in the label
;  469          0850  5
;  470          0851  5               SCRATCH[VL1$B_VOLACCESS] = .CHAR[0];
;  471          0852  5
;  472          0853  5               ! fill in the VOL2 VMS owner field
;  473          0854  5
;  474          0855  5               IF .ORB[ORB$V_PROT_16]
;  475          0856  5               THEN TMP_PROT = .ORB[ORB$W_PROT]
;  476          0857  5               ELSE
;  477          0858  6                   BEGIN
;  478          0859  6                   TMP_PROT<0,4> = .(ORB[ORB$L_SYS_PROT])<0,4>;
;  479          0860  6                   TMP_PROT<4,4> = .(ORB[ORB$L_OWN_PROT])<0,4>;
;  480          0861  6                   TMP_PROT<8,4> = .(ORB[ORB$L_GRP_PROT])<0,4>;
;  481          0862  6                   TMP_PROT<12,4> = .(ORB[ORB$[_WOR_PROT])<0,4>;
;  482          0863  5                   END;
;  483          0864  5               FORMAT_VOLOWNER(SCRATCH2, .ORB[ORB$L_OWNER], .TMP_PROT);
;  484          0865  5
;  485          0866  5               ! If a VMS proctection is specified and the user does not
;  486          0867  5               ! wish us to limit this to only ANSI standard only then
;  487          0868  5               ! write our system code in the VOL1 label. This will
;  488          0869  5               ! tell other implemenations that the VOL2 label on this
;  489          0870  5               ! tape was written by VMS.
;  490          0871  5
;  491          0872  6               IF NOT (.CURRENT_VCB[VCB$V_INTCHG]
```

```
;  492          0873  6                    AND .CURRENT_VCB [VCB$V_NOVOL2])
;  493          0874  6                    AND (.ORB[ORB$L_SYS_PROT] NEQ 0 OR
;  494          0875  6                         .ORB[ORB$L_OWN_PROT] NEQ 0 OR
;  495          0876  6                         .ORB[ORB$L_GRP_PROT] NEQ 0 OR
;  496          0877  6                         .ORB[ORB$L_WOR_PROT] NEQ 0)
;  497          0878  5                THEN
;  498          0879  6                BEGIN
;  499          0880  6                     CH$MOVE(10,STARID,SCRATCH[VL1$T_SYSCODE]);
;  500          0881  6                     SCRATCH[VL1$B_LBLSTDVER] = '4';
;  501          0882  5                END;
;  502          0883  5
;  503          0884  5                ! set the same characteristics and if that succeeds write the
;  504          0885  5                ! label.
;  505          0886  5
;  506          0887  5                IF ISSUE_IO(IO$_SETMODE, SAVE_DEVCHAR, 0)
;  507          0888  5                   THEN
;  508          0889  5                        STATUS = ISSUE_IO(IO$_WRITELBLK, .SCRATCH, ANSI_LBLSZ);
;  509          0890  5
;  510          0891  5             ! If the frist write worked, then check to see if a VOL2 label needs
;  511          0892  5             ! to be written. If it does and that worked then exitloop.
;  512          0893  5
;  513          0894  5                IF .STATUS
;  514          0895  5                   THEN
;  515          0896  6                   BEGIN
;  516          0897  7                        IF NOT (.CURRENT_VCB[VCB$V_INTCHG]
;  517          0898  7                              AND .CURRENT_VCB [VCB$V_NOVOL2])
;  518          0899  7                              AND (.ORB[ORB$L_SYS_PROT] NEQ 0 OR
;  519          0900  7                                   .ORB[ORB$L_OWN_PROT] NEQ 0 OR
;  520          0901  7                                   .ORB[ORB$L_GRP_PROT] NEQ 0 OR
;  521          0902  7                                   .ORB[ORB$L_WOR_PROT] NEQ 0)
;  522          0903  6                           THEN
;  523          0904  6                                STATUS = ISSUE_IO (IO$_WRITELBLK, SCRATCH2,
;  524          0905  6                                                   ANSI_LBLSZ);
;  525          0906  6                             IF .STATUS THEN EXITLOOP;
;  526          0907  5                        END;
;  527          0908  5
;  528          0909  5
;  529          0910  5                    IF .STATUS<0,16> EQL SS$_WRITLCK THEN ERROR_NO = MOUN$_WRITLCK;
;  530          0911  5
;  531          0912  5                END
;  532          0913  4                ELSE
;  533          0914  4                    ERROR_NO = MOUN$_TAPEPOSLOST;
;  534          0915  3           END;
;  535          0916  3           PRINT_OPR_MSG(.ERROR_NO, 0,
;  536          0917  3                         .CVT_DEVNAM_LENGTH,CVT_DEVNAM);
;  537          0918  3
;  538          0919  3           ! force physical mount
;  539          0920  3
;  540          0921  3           FLAGS = $FIELDMASK(MOU$V_REWIND) + $FIELDMASK(MOU$V_MOUNTERR);
;  541          0922  3           KERNEL_CALL(RESET_UNIT);
;  542          0923  2           END;
;  543          0924  2
;  544          0925  1     END;                                        ! end of routine


                                                  000DB          .BLKB   1
```

```
     00 00 41 31 31 45 4C 49 46 43 45 44 000DC P.AAA:  .ASCII  \DECFILE11A\<0><0>              ;

                                                  STARID=          P.AAA
                                                           .EXTRN  SYS$MTACCESS

                        5E      FF68  CE  9E 00000 GTNEXT_VOL_WRIT::
                                                           MOVAB   -152(SP), SP                ; 0645
                                      7E  D4 00005         CLRL    -(SP)                       ; 0719
                                      5E  DD 00007         PUSHL   SP                          ;
                              0000V   CF  9F 00009         PUSHAB  INC_VOL_SECTION             ;
                 00000000G  9F        03  FB 0000D         CALLS   #3, @#SYS$CMKRNL            ;
                            50  0000G CF  D0 00014         MOVL    CURRENT_UCB, R0             ; 0720
                   0080     CE    40  A0  7D 00019         MOVQ    64(R0), SAVE_DEVCHAR        ;
           59      0000G    CF 00000140 8F C1 0001F        ADDL3   #320, HDR1, SCRATCH         ; 0722
                     18     AE        01  D0 00029         MOVL    #1, FLAGS                   ; 0723
                              0088    CE  9F 0002D         PUSHAB  CVT_DEVNAM                  ; 0728
                                20    AE  9F 00031         PUSHAB  CVT_DEVNAM_LENGTH           ;
                   0000G    CF        02  FB 00034         CALLS   #2, GET_DEV_NAME            ;
                     14     AE    1C  AE  9A 00039         MOVZBL  CVT_DEVNAM_LENGTH, 20(SP)   ; 0773
                     14     AE    1C  AE  9A 0003E         MOVZBL  CVT_DEVNAM_LENGTH, 20(SP)   ; 0917
                                  18  AE  DD 00043 1$:     PUSHL   FLAGS                       ; 0740
                            7E    2F  AB  9A 00046         MOVZBL  47(CURRENT_VCB), -(SP)      ;
                   0000G    CF        02  FB 0004A         CALLS   #2, MOUNT_VOL               ;
                     08     AE        50  D0 0004F         MOVL    R0, MVL_ENTRY               ;
                            57    34  AB  D0 00053         MOVL    52(CURRENT_VCB), MVL        ; 0741
                                      50  D4 00057         CLRL    R0                          ; 0747
                   81E3     8F  0000G CF  B1 00059         CMPW    MAIL+2, #33251              ;
                                      02  12 00060         BNEQ    2$                          ;
                                      50  D6 00062         INCL    R0                          ;
     5A            01               00    50  F0 00064 2$:     INSV    R0, #0, #1, OPR_FLAG        ; 0746
                                      50  D4 00069         CLRL    R0                          ; 0749
                   81D3     8F  0000G CF  B1 0006B         CMPW    MAIL+2, #33235              ;
                                      02  12 00072         BNEQ    3$                          ;
                                      50  D6 00074         INCL    R0                          ;
     5A            01               01    50  F0 00076 3$:     INSV    R0, #1, #1, OPR_FLAG        ; 0748
                                    43    5A  E8 0007B         BLBS    OPR_FLAG, 6$                ; 0754
                   3E        2D     AB    02  E0 0007E         BBS     #2, 45(CURRENT_VCB), 6$     ;
                             0C     AE    0A  A9  90 00083     MOVB    10(SCRATCH), ACCESS_CHAR    ; 0759
              70   AE        25     A9    0E  28 00088         MOVC3   #14, 37(SCRATCH), VOL_OWNER ; 0760
     50            5A               01    01  EF 0008E         EXTZV   #1, #1, OPR_FLAG, R0        ; 0762
     51       2D   AB               01    03  EF 00093         EXTZV   #3, #1, 45(CURRENT_VCB), R1 ;
                   7E               50    51  C9 00099         BISL3   R1, R0, -(SP)               ;
                                  0C  AE  DD 0009D         PUSHL   MVL_ENTRY                   ; 0761
                                   0000V  30 000A0         BSBW    CHECK_HDR                   ;
                            5E        08  C0 000A3         ADDL2   #8, SP                      ;
                            58        50  D0 000A6         MOVL    R0, ERROR_NO                ;
                            15        58  E8 000A9         BLBS    ERROR_NO, 6$                ; 0766
                   08       5A        01  E0 000AC         BBS     #1, OPR_FLAG, 5$            ;
                   03        2D     AB    03  E0 000B0     BBS     #3, 45(CURRENT_VCB), 5$     ;
                                    019F  31 000B5 4$:     BRW     22$                         ;
                 007280FC   8F        58  D1 000B8 5$:     CMPL    ERROR_NO, #7504124          ; 0767
                                      F4  12 000BF         BNEQ    4$                          ;
                       58 00728124    8F  D0 000C1 6$:     MOVL    #7504164, ERROR_NO          ; 0781
                   0000G    CF        00  FB 000C8         CALLS   #0, REWIND_AND_WAIT         ; 0785
                            E5        50  E9 000CD         BLBC    R0, 4$                      ;
  0050     8F      20       6E        00  2C 000D0         MOVC5   #0, (SP), #32, #80, (SCRATCH) ; 0791
                                      69     000D7                                             ;
```

```
0050  8F      20          6E          00  2C 000D8          MOVC5   #0, (SP), #32, #80, SCRATCH2          ; 0792
                              20      AE      000DF
                        69 314C4F56   8F  D0 000E1          MOVL    #827084630, (SCRATCH)                 ; 0796
                4F  A9      22  A7      30  81 000E8          ADDB3   #48, 34(MVL), 79(SCRATCH)             ; 0797
                            20  AE 324C4F56  8F  D0 000EE    MOVL    #843861846, SCRATCH2                  ; 0798
                            24  AE 20432544  8F  D0 000F6    MOVL    #541271364, SCRATCH2+4                ; 0799
                04  A9      08  BE          06  28 000FE    MOVC3   #6, @MVL_ENTRY, 4(SCRATCH)            ; 0804
                                    0000G   CF  D5 00104    TSTL    MAILSZ                                ; 0813
                0D              5A          01  E0 0010A    BBS     #1, OPR_FLAG, 7$                      ;
                                0A          5A  E8 0010E    BLBS    OPR_FLAG, 7$                          ;
                05      2D      AB          03  E0 00111    BBS     #3, 45(CURRENT_VCB), 7$               ; 0814
                0D      2D      AB          02  E1 00116    BBC     #2, 45(CURRENT_VCB), 8$               ; 0815
                        0C      AE      12  A7  90 0011B 7$: MOVB    18(MVL), ACCESS_CHAR                  ; 0818
                25  A9      14  A7          0E  28 00120    MOVC3   #14, 20(MVL), 37(SCRATCH)             ; 0820
                                            06  11 00126    BRB     9$                                    ; 0813
                25  A9      70  AE          0E  28 00128 8$: MOVC3   #14, VOL_OWNER, 37(SCRATCH)          ; 0825
                            50      0000G   CF  D0 0012E 9$: MOVL    CURRENT_UCB, R0                       ; 0834
                            56      1C      A0  D0 00133    MOVL    28(R0), ORB                           ;
                                            50  DD 00137    PUSHL   R0                                    ; 0835
                                            01  DD 00139    PUSHL   #1                                    ;
                                            5E  DD 0013B    PUSHL   SP                                    ;
                                    0000G   CF  9F 0013D    PUSHAB  GET_RECORD                            ;
                    00000000G       9F          04  FB 00141    CALLS   #4, @#SYS$CMKRNL                  ;
                            04      AE          50  D0 00148    MOVL    R0, CURRENT_RECORD                ;
                                            02  DD 0014C    PUSHL   #2                                    ; 0841
                                            01  DD 0014E    PUSHL   #1                                    ;
                                7E      14  AE  9A 00150    MOVZBL  ACCESS_CHAR, -(SP)                    ;
                                7E      22  A7  9A 00154    MOVZBL  34(MVL), -(SP)                        ;
                                            66  DD 00158    PUSHL   (ORB)                                 ;
                                            7E  D4 0015A    CLRL    -(SP)                                 ;
                    00000000G       00          06  FB 0015C    CALLS   #6, SYS$MTACCESS                  ;
                                6E          50  D0 00163    MOVL    R0, CHAR                              ;
                                        0000G   CF  DD 00166    PUSHL   CURRENT_UCB                       ; 0843
                                            01  DD 0016A    PUSHL   #1                                    ;
                                            5E  DD 0016C    PUSHL   SP                                    ;
                                    0000G   CF  9F 0016E    PUSHAB  GET_RECORD                            ;
                    00000000G       9F          04  FB 00172    CALLS   #4, @#SYS$CMKRNL                  ;
                            10      AE          50  D0 00179    MOVL    R0, STATUS                        ;
                            10      AE      04  AE  D1 0017D    CMPL    CURRENT_RECORD, STATUS            ; 0844
                                            03  13 00182    BEQL    10$                                   ;
                                        00C9    31 00184    BRW     21$                                   ;
                            0A      A9          6E  90 00187 10$: MOVB    CHAR, 10(SCRATCH)                 ; 0851
                                06      0B      A6  E9 0018B    BLBC    11(ORB), 11$                      ; 0855
                                50      18      A6  B0 0018F    MOVW    24(ORB), TMP_PROT                 ; 0856
                                            18  11 00193    BRB     12$                                   ;
50              04              00      18      A6  F0 00195 11$: INSV    24(ORB), #0, #4, TMP_PROT         ; 0859
50              04              04      1C      A6  F0 0019B    INSV    28(ORB), #4, #4, TMP_PROT         ; 0860
50              04              08      20      A6  F0 001A1    INSV    32(ORB), #8, #4, TMP_PROT         ; 0861
50              04              0C      24      A6  F0 001A7    INSV    36(ORB), #12, #4, TMP_PROT        ; 0862
                                7E          50  3C 001AD 12$: MOVZWL  TMP_PROT, -(SP)                   ; 0864
                                            66  DD 001B0    PUSHL   (ORB)                                 ;
                                        28  AE  9F 001B2    PUSHAB  SCRATCH2                              ;
                            0000G       CF      03  FB 001B5    CALLS   #3, FORMAT_VOLOWNER               ;
                        05      2C      AB          04  E1 001BA    BBC     #4, 44(CURRENT_VCB), 13$      ; 0872
                        1F      2C      AB          06  E0 001BF    BBS     #6, 44(CURRENT_VCB), 15$      ; 0873
                                        18      A6  D5 001C4 13$: TSTL    24(ORB)                           ; 0874
```

```
                                  0F  12 001C7        BNEQ    14$                                ;
                          1C      A6  D5 001C9        TSTL    28(ORB)                            ; 0875
                                  0A  12 001CC        BNEQ    14$                                ;
                          20      A6  D5 001CE        TSTL    32(ORB)                            ; 0876
                                  05  12 001D1        BNEQ    14$                                ;
                          24      A6  D5 001D3        TSTL    36(ORB)                            ; 0877
                                  0B  13 001D6        BEQL    15$                                ;
  18    A9      FE17    CF        0A  28 001D8 14$:   MOVC3   #10, STARID, 24(SCRATCH)           ; 0880
                  4F    A9        34  90 001DF        MOVB    #52, 79(SCRATCH)                   ; 0881
                                  7E  D4 001E3 15$:   CLRL    -(SP)                              ; 0887
                        0084      CE  9F 001E5        PUSHAB  SAVE_DEVCHAR                       ;
                                  23  DD 001E9        PUSHL   #35                                ;
                               0000G  30 001EB        BSBW    ISSUE_IO                           ;
                        5E        0C  C0 001EE        ADDL2   #12, SP                            ;
                        12        50  E9 001F1        BLBC    R0, 16$                            ;
                        7E    50  8F  9A 001F4        MOVZBL  #80, -(SP)                         ; 0889
                                  59  DD 001F8        PUSHL   SCRATCH                            ;
                                  20  DD 001FA        PUSHL   #32                                ;
                               0000G  30 001FC        BSBW    ISSUE_IO                           ;
                        5E        0C  C0 001FF        ADDL2   #12, SP                            ;
                  10    AE        50  D0 00202        MOVL    R0, STATUS                         ;
                        35    10  AE  E9 00206 16$:   BLBC    STATUS, 20$                        ; 0894
        05        2C    AB        04  E1 0020A        BBC     #4, 44(CURRENT_VCB), 17$           ; 0897
        27        2C    AB        06  E0 0020F        BBS     #6, 44(CURRENT_VCB), 19$           ; 0898
                          18      A6  D5 00214 17$:   TSTL    24(ORB)                            ; 0899
                                  0F  12 00217        BNEQ    18$                                ;
                          1C      A6  D5 00219        TSTL    28(ORB)                            ; 0900
                                  0A  12 0021C        BNEQ    18$                                ;
                          20      A6  D5 0021E        TSTL    32(ORB)                            ; 0901
                                  05  12 00221        BNEQ    18$                                ;
                          24      A6  D5 00223        TSTL    36(ORB)                            ; 0902
                                  13  13 00226        BEQL    19$                                ;
                        7E    50  8F  9A 00228 18$:   MOVZBL  #80, -(SP)                         ; 0904
                              24  AE  9F 0022C        PUSHAB  SCRATCH2                           ;
                                  20  DD 0022F        PUSHL   #32                                ;
                               0000G  30 00231        BSBW    ISSUE_IO                           ;
                        5E        0C  C0 00234        ADDL2   #12, SP                            ;
                  10    AE        50  D0 00237        MOVL    R0, STATUS                         ;
                        3F    10  AE  E8 0023B 19$:   BLBS    STATUS, 23$                        ; 0906
                025C    8F    10  AE  B1 0023F 20$:   CMPW    STATUS, #604                       ; 0910
                                  10  12 00245        BNEQ    22$                                ;
                  58 00728134     8F  D0 00247        MOVL    #7504180, ERROR_NO                 ;
                                  07  11 0024E        BRB     22$                                ; 0844
                  58 00728274     8F  D0 00250 21$:   MOVL    #7504500, ERROR_NO                 ; 0914
                        0088      CE  9F 00257 22$:   PUSHAB  CVT_DEVNAM                         ; 0916
                          18      AE  DD 0025B        PUSHL   24(SP)                             ; 0917
                                  7E  D4 0025E        CLRL    -(SP)                              ; 0916
                                  58  DD 00260        PUSHL   ERROR_NO                           ;
                               0000G  30 00262        BSBW    PRINT_OPR_MSG                      ;
                        5E        10  C0 00265        ADDL2   #16, SP                            ;
                  18    AE        09  D0 00268        MOVL    #9, FLAGS                          ; 0921
                                  7E  D4 0026C        CLRL    -(SP)                              ; 0922
                                  5E  DD 0026E        PUSHL   SP                                 ;
                        0000V     CF  9F 00270        PUSHAB  RESET_UNIT                         ;
         00000000G      9F        03  FB 00274        CALLS   #3, @#SYS$CMKRNL                   ;
                               FDC5   31 0027B        BRW     1$                                 ; 0730
                        5E  0098  CE  9E 0027E 23$:   MOVAB   152(SP), SP                        ; 0925
```

```
                                        05 00283        RSB                                     ;

; Routine Size:  644 bytes,    Routine Base:  $CODE$ + 00E8

;  545          0926  1
```

```
;  547          0927  1 ROUTINE INC_VOL_SECTION : COMMON_CALL NOVALUE =
;  548          0928  1
;  549          0929  1 !++
;  550          0930  1 !
;  551          0931  1 ! FUNCTIONAL DESCRIPTION:
;  552          0932  1 !       This routine increments the relative volume number
;  553          0933  1 !       and the file section number
;  554          0934  1 !
;  555          0935  1 ! CALLING SEQUENCE:
;  556          0936  1 !       INC_VOL_SECTION(), CALLED IN KERNEL MODE
;  557          0937  1 !
;  558          0938  1 ! INPUT PARAMETERS:
;  559          0939  1 !       NONE
;  560          0940  1 !
;  561          0941  1 ! IMPLICIT INPUTS:
;  562          0942  1 !       CURRENT_VCB - address of volume control block
;  563          0943  1 !
;  564          0944  1 ! OUTPUT PARAMETERS:
;  565          0945  1 !       NONE
;  566          0946  1 !
;  567          0947  1 ! IMPLICIT OUTPUTS:
;  568          0948  1 !       CURRENT_VCB[VCB$B_CUR_RVN] incremented
;  569          0949  1 !       CURRENT_VCB[VCB$W_CUR_SEQ] incremented
;  570          0950  1 !
;  571          0951  1 ! ROUTINE VALUE:
;  572          0952  1 !       NONE
;  573          0953  1 !
;  574          0954  1 ! SIDE EFFECTS:
;  575          0955  1 !       NONE
;  576          0956  1 !
;  577          0957  1 !--
;  578          0958  1
;  579          0959  2     BEGIN
;  580          0960  2
;  581          0961  2     EXTERNAL REGISTER
;  582          0962  2         COMMON_REG;
;  583          0963  2
;  584          0964  2     CURRENT_VCB[VCB$B_CUR_RVN] = .CURRENT_VCB[VCB$B_CUR_RVN] + 1;
;  585          0965  2     CURRENT_VCB[VCB$W_CUR_SEQ] = .CURRENT_VCB[VCB$W_CUR_SEQ] + 1;
;  586          0966  2     CURRENT_VCB[VCB$B_TM] = 0;
;  587          0967  2     CURRENT_VCB[VCB$L_ST_RECORD] = 0;
;  588          0968  1     END;                                    ! end of routine


                                    0000 00000 INC_VOL_SECTION:
                                                .WORD   Save nothing                    ; 0927
                                2F  AB  96 00002        INCB    47(CURRENT_VCB)         ; 0964
                                26  AB  B6 00005        INCW    38(CURRENT_VCB)         ; 0965
                                2E  AB  94 00008        CLRB    46(CURRENT_VCB)         ; 0966
                                30  AB  D4 0000B        CLRL    48(CURRENT_VCB)         ; 0967
                                        04 0000E        RET                             ; 0968

; Routine Size:  15 bytes,    Routine Base:  $CODE$ + 036C
```

```
;  590          0969  1 ROUTINE UPDATE_MVL_LBL (MVL_ENTRY, ADDR) : COMMON_CALL NOVALUE =
;  591          0970  1
;  592          0971  1 !++
;  593          0972  1 !
;  594          0973  1 ! FUNCTIONAL DESCRIPTION:
;  595          0974  1 !       This routine updates the relative volume label from the vol1 label
;  596          0975  1 !
;  597          0976  1 ! CALLING SEQUENCE:
;  598          0977  1 !       UPDATE_MVL_LBL(ARG1,ARG2)
;  599          0978  1 !
;  600          0979  1 ! INPUT PARAMETERS:
;  601          0980  1 !       ARG1 - address of mvl entry for current volume
;  602          0981  1 !       ARG2 - address of volume label on this tape volume
;  603          0982  1 !
;  604          0983  1 ! IMPLICIT INPUTS:
;  605          0984  1 !       NONE
;  606          0985  1 !
;  607          0986  1 ! OUTPUT PARAMETERS:
;  608          0987  1 !       NONE
;  609          0988  1 !
;  610          0989  1 ! IMPLICIT OUTPUTS:
;  611          0990  1 !       NONE
;  612          0991  1 !
;  613          0992  1 ! ROUTINE VALUE:
;  614          0993  1 !       NONE
;  615          0994  1 !
;  616          0995  1 ! SIDE EFFECTS:
;  617          0996  1 !       NONE
;  618          0997  1 !
;  619          0998  1 ! USER ERRORS:
;  620          0999  1 !       NONE
;  621          1000  1 !
;  622          1001  1 !--
;  623          1002  1
;  624          1003  2     BEGIN
;  625          1004  2
;  626          1005  2     EXTERNAL REGISTER COMMON_REG;
;  627          1006  2
;  628          1007  2     EXTERNAL
;  629          1008  2         ANSI_A_GOOD  : VECTOR [ , BYTE ];! translation table for ANSI 'a' char
;  630          1009  2
;  631          1010  2     MAP
;  632          1011  2         MVL_ENTRY    : REF BBLOCK;
;  633          1012  2
;  634          1013  2     ! translate the label into upper case and put in ' ' for any non-ANSI
;  635          1014  2     !  'a' characters found
;  636          1015  2
;  637          1016  2     CH$TRANSLATE (ANSI_A_GOOD, VL1$S_VOLLBL, .ADDR, ' ',
;  638          1017  2                                MVL$S_VOLLBL, MVL_ENTRY [MVL$T_VOLLBL] );
;  639          1018  2     MVL_ENTRY [ MVL$V_UNUSED ] = 0;
;  640          1019  1     END;


                                                        .EXTRN  ANSI_A_GOOD

                                    007C 00000 UPDATE_MVL_LBL:
```

```
                                                .WORD   Save R2,R3,R4,R5,R6                             ; 0969
                        56      04      AC  D0 00002    MOVL    MVL_ENTRY, R6                           ; 1017
    0000G  CF       20      08      BC  2E 00006    MOVTC   #6, @ADDR, #32, ANSI_A_GOOD, #6, (R6)   ;
                            66          06      0000E                                                   ;
                        07      A6          02  8A 00010    BICB2   #2, 7(R6)                       ; 1018
                                            04  00014    RET                                         ; 1019

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 037B
```

```
;  642          1020  1 ROUTINE CHECK_HDR ( MVL_ENTRY, SLASH_INIT ) : L$CHECK_HDR =
;  643          1021  1
;  644          1022  1 !++
;  645          1023  1 !
;  646          1024  1 ! FUNCTIONAL DESCRIPTION:
;  647          1025  1 !       This routine checks that the tape can be overwritten.
;  648          1026  1 !
;  649          1027  1 ! CALLING SEQUENCE:
;  650          1028  1 !       CHECK_HDR(ARG1,ARG2)
;  651          1029  1 !
;  652          1030  1 ! INPUT PARAMETERS:
;  653          1031  1 !       ARG1 - address of current mounted volume entry
;  654          1032  1 !       ARG2 - is this a "/INIT"
;  655          1033  1 !
;  656          1034  1 ! IMPLICIT INPUTS:
;  657          1035  1 !       NONE
;  658          1036  1 !
;  659          1037  1 ! OUTPUT PARAMETERS:
;  660          1038  1 !       NONE
;  661          1039  1 !
;  662          1040  1 ! IMPLICIT OUTPUTS:
;  663          1041  1 !       NONE
;  664          1042  1 !
;  665          1043  1 ! ROUTINE VALUE:
;  666          1044  1 !       1 - ok to write
;  667          1045  1 !       various error codes
;  668          1046  1 !
;  669          1047  1 ! SIDE EFFECTS:
;  670          1048  1 !       NONE
;  671          1049  1 !
;  672          1050  1 !--
;  673          1051  1
;  674          1052  2     BEGIN
;  675          1053  2
;  676          1054  2     MAP
;  677          1055  2         MVL_ENTRY       : REF BBLOCK;
;  678          1056  2
;  679          1057  2     EXTERNAL REGISTER
;  680          1058  2         SCRATCH = 9     : REF BBLOCK,
;  681          1059  2         COMMON_REG;
;  682          1060  2
;  683          1061  2     BIND
;  684          1062  2         USER_VOL_LABEL  = UPLIT ( 'UVL' ),      ! user's volume labels code
;  685          1063  2         VOLUME_LABEL    = UPLIT ( 'VOL' );      ! other volume labels code
;  686          1064  2
;  687          1065  2     LOCAL
;  688          1066  2         MVL     : REF BBLOCK,                   ! MVL address
;  689          1067  2         ORB     : REF BBLOCK,                   ! ORB address
;  690          1068  2         STATUS,
;  691          1069  2         CURRENT_RECORD,                         ! curr record drive is reading
;  692          1070  2         ACCESS;                                 ! Users' access to overwrite tape
;  693          1071  2
;  694          1072  2     !  loop till we find HDR1
;  695          1073  2
;  696          1074  2     WHILE 1
;  697          1075  2     DO
;  698          1076  3         BEGIN
```

```
;  699          1077  3
;  700          1078  3          STATUS = ISSUE_IO(IO$_READLBLK, .SCRATCH, ANSI_LBLSZ);
;  701          1079  3
;  702          1080  3          IF (.STATUS<0,16> EQL SS$_ENDOFFILE) AND .SLASH_INIT
;  703          1081  3          THEN RETURN TRUE;
;  704          1082  3
;  705          1083  3          IF (NOT .STATUS) AND (.STATUS<0,16> NEQ SS$_DATAOVERUN)
;  706          1084  3          THEN RETURN MOUN$_IOERROR;
;  707          1085  3
;  708          1086  3          IF .(.SCRATCH) EQL 'HDR1' THEN EXITLOOP;
;  709          1087  3
;  710          1088  3          ! if we do not see a valid member of the volume label group THEN FAIL
;  711          1089  3
;  712          1090  3          IF NOT (    ( CH$EQL ( 3, .SCRATCH, 3, USER_VOL_LABEL ))
;  713          1091  3                   OR ( CH$EQL ( 3, .SCRATCH, 3, VOLUME_LABEL ))
;  714          1092  3                 )
;  715          1093  3          THEN RETURN MOUN$_NOTANSI;
;  716          1094  3
;  717          1095  2          END;
;  718          1096  2
;  719          1097  2      ! Call the accessibility system service to check the accessibility char
;  720          1098  2      ! on the HDR1 label.
;  721          1099  2      ! First keep the record that the UCB is reading. The accessibility
;  722          1100  2      ! routine can not move the tape from under us! Thus we will compare
;  723          1101  2      ! this to the field after the call and if the tape was moved we punt
;  724          1102  2      ! the operation. The check the code return from the system service
;  725          1103  2      ! to determine what type of access the user was granted.
;  726          1104  2
;  727          1105  2      MVL = .CURRENT_VCB[VCB$L_MVL];
;  728          1106  2      ORB = .CURRENT_UCB[UCB$L_ORB];
;  729          1107  2      CURRENT_RECORD = KERNEL_CALL(GET_RECORD, .CURRENT_UCB);
;  730        P 1108  2      ACCESS = $MTACCESS(LBLNAM = .SCRATCH,
;  731        P 1109  2                         UIC = .ORB[ORB$L_OWNER],
;  732        P 1110  2                         STD_VERSION = .MVL[MVL$B_STDVER],
;  733        P 1111  2                         ACCESS_CHAR = 0,
;  734        P 1112  2                         ACCESS_SPEC = MTA$K_NOCHAR,
;  735          1113  2                         TYPE = MTA$K_INHDR1);
;  736          1114  2      STATUS = KERNEL_CALL(GET_RECORD, .CURRENT_UCB);
;  737          1115  2      IF .CURRENT_RECORD NEQ .STATUS
;  738          1116  2           THEN RETURN (MOUN$_TAPEPOSLOST);
;  739          1117  2
;  740          1118  2       IF .ACCESS EQL SS$_FILACCERR
;  741          1119  2           THEN
;  742          1120  3           BEGIN
;  743          1121  4              IF NOT (.CURRENT_VCB[VCB$V_OVRACC] AND .MVL_ENTRY [ MVL$V_OVERIDE ])
;  744          1122  3                   THEN RETURN MOUN$_ACCERR;
;  745          1123  3              ACCESS = SS$_NORMAL;
;  746          1124  2           END;
;  747          1125  2
;  748          1126  2       IF .ACCESS EQL SS$_NOVOLACC
;  749          1127  2           THEN RETURN MOUN$_NOVOLACC;
;  750          1128  2
;  751          1129  2       IF .ACCESS EQL SS$_NOFILACC
;  752          1130  2           THEN RETURN MOUN$_NOFILACC;
;  753          1131  2
;  754          1132  4       IF NOT (    (.CURRENT_VCB[VCB$V_OVREXP] AND .MVL_ENTRY [ MVL$V_OVERIDE ])
;  755          1133  3                OR
```

```
;  756          1134  3                EXPIRED ( SCRATCH[HD1$T_EXPIREDT] )
;  757          1135  3            )
;  758          1136  2    THEN RETURN  MOUN$_FILNOTEXP;
;  759          1137  2
;  760          1138  2    RETURN TRUE;
;  761          1139  2
;  762          1140  1    END;                                    ! end of routine CHECK_HDR


                                00  4C  56  55  00390 P.AAB:  .ASCII  \UVL\<0>                          ;
                                00  4C  4F  56  00394 P.AAC:  .ASCII  \VOL\<0>                          ;

                                                    USER_VOL_LABEL=     P.AAB
                                                    VOLUME_LABEL=       P.AAC


                                    3C  BB 00000 CHECK_HDR:
                                                        PUSHR   #^M<R2,R3,R4,R5>                        ; 1020
                          7E        50   8F  9A 00002 1$:     MOVZBL  #80, -(SP)                        ; 1078
                                      59  DD 00006            PUSHL   SCRATCH                           ;
                                      21  DD 00008            PUSHL   #33                               ;
                                   0000G 30 0000A             BSBW    ISSUE_IO                          ;
                          5E          0C  C0 0000D            ADDL2   #12, SP                           ;
                          54          50  D0 00010            MOVL    R0, STATUS                        ;
                    0870  8F          54  B1 00013            CMPW    STATUS, #2160                     ; 1080
                                      07  12 00018            BNEQ    2$                                ;
                          03        18  AE  E9 0001A            BLBC    SLASH_INIT, 2$                    ;
                                   00F6 31 0001E              BRW     14$                               ;
                          10          54  E8 00021 2$:        BLBS    STATUS, 3$                        ; 1083
                        0838  8F      54  B1 00024            CMPW    STATUS, #2104                     ;
                                      09  13 00029            BEQL    3$                                ;
                              50 00728124 8F  D0 0002B        MOVL    #7504164, R0                      ; 1084
                                      79  11 00032            BRB     5$                                ;
                    31524448  8F      69  D1 00034 3$:        CMPL    (SCRATCH), #827475016             ; 1086
                                      17  13 0003B            BEQL    4$                                ;
              B6    AF                69  03 29 0003D        CMPC3   #3, (SCRATCH), USER_VOL_LABEL     ; 1090
                                      BE  13 00042            BEQL    1$                                ;
              B3    AF                69  03 29 00044        CMPC3   #3, (SCRATCH), VOLUME_LABEL       ; 1091
                                      B7  13 00049            BEQL    1$                                ;
                              50 007280FC 8F  D0 0004B        MOVL    #7504124, R0                      ; 1093
                                      79  11 00052            BRB     8$                                ;
                          52        34  AB  D0 00054 4$:        MOVL    52(CURRENT_VCB), MVL              ; 1105
                          50      0000G CF  D0 00058            MOVL    CURRENT_UCB, R0                   ; 1106
                          53        1C  A0  D0 0005D            MOVL    28(R0), ORB                       ;
                                      50  DD 00061            PUSHL   R0                                ; 1107
                                      01  DD 00063            PUSHL   #1                                ;
                                      5E  DD 00065            PUSHL   SP                                ;
                                   0000G CF  9F 00067         PUSHAB  GET_RECORD                        ;
                    00000000G     9F      04  FB 0006B        CALLS   #4, @#SYS$CMKRNL                  ;
                                  55      50  D0 00072        MOVL    R0, CURRENT_RECORD                ;
                                      01  DD 00075            PUSHL   #1                                ; 1113
                                      7E  7C 00077            CLRQ    -(SP)                             ;
                          7E        22  A2  9A 00079        MOVZBL  34(MVL), -(SP)                    ;
                                      63  DD 0007D            PUSHL   (ORB)                             ;
                                      59  DD 0007F            PUSHL   SCRATCH                           ;
                    00000000G     00      06  FB 00081        CALLS   #6, SYS$MTACCESS                  ;
```

```
                    53               50 D0 00088          MOVL    R0, ACCESS                     ;
                           0000G     CF DD 0008B          PUSHL   CURRENT_UCB                    ; 1114
                                     01 DD 0008F          PUSHL   #1                             ;
                                     5E DD 00091          PUSHL   SP                             ;
                           0000G     CF 9F 00093          PUSHAB  GET_RECORD                     ;
           00000000G  9F             04 FB 00097          CALLS   #4, @#SYS$CMKRNL               ;
                    54               50 D0 0009E          MOVL    R0, STATUS                     ;
                    54               55 D1 000A1          CMPL    CURRENT_RECORD, STATUS         ; 1115
                                     09 13 000A4          BEQL    6$                             ;
                 50 00728274         8F D0 000A6          MOVL    #7504500, R0                   ; 1116
                                     6B 11 000AD 5$:      BRB     15$                            ;
           0000009C   8F             53 D1 000AF 6$:      CMPL    ACCESS, #156                   ; 1118
                                     1A 12 000B6          BNEQ    10$                            ;
    09          2C    AB             01 E1 000B8          BBC     #1, 44(CURRENT_VCB), 7$        ; 1121
                      50        14   AE D0 000BD          MOVL    MVL_ENTRY, R0                  ;
    09          07    A0             02 E0 000C1          BBS     #2, 7(R0), 9$                  ;
                 50 007280E4         8F D0 000C6 7$:      MOVL    #7504100, R0                   ; 1122
                                     4B 11 000CD 8$:      BRB     15$                            ;
                    53               01 D0 000CF 9$:      MOVL    #1, ACCESS                     ; 1123
           000022A4   8F             53 D1 000D2 10$:     CMPL    ACCESS, #8868                  ; 1126
                                     09 12 000D9          BNEQ    11$                            ;
                 50 00728264         8F D0 000DB          MOVL    #7504484, R0                   ; 1127
                                     36 11 000E2          BRB     15$                            ;
           000022AC   8F             53 D1 000E4 11$:     CMPL    ACCESS, #8876                  ; 1129
                                     09 12 000EB          BNEQ    12$                            ;
                 50 0072826C         8F D0 000ED          MOVL    #7504492, R0                   ; 1130
                                     24 11 000F4          BRB     15$                            ;
                      09        2C   AB E9 000F6 12$:     BLBC    44(CURRENT_VCB), 13$           ; 1132
                      50        14   AE D0 000FA          MOVL    MVL_ENTRY, R0                  ;
    14          07    A0             02 E0 000FE          BBS     #2, 7(R0), 14$                 ;
                                2F   A9 9F 00103 13$:     PUSHAB  47(SCRATCH)                    ; 1134
              0000G   CF             01 FB 00106          CALLS   #1, EXPIRED                    ;
                      09             50 E8 0010B          BLBS    R0, 14$                        ;
                 50 007280EC         8F D0 0010E          MOVL    #7504108, R0                   ; 1136
                                     03 11 00115          BRB     15$                            ;
                    50               01 D0 00117 14$:     MOVL    #1, R0                         ; 1138
                                     3C BA 0011A 15$:     POPR    #^M<R2,R3,R4,R5>               ; 1140
                                     05 0011C             RSB                                    ;

; Routine Size:  285 bytes,    Routine Base:  $CODE$ + 0398


;  763          1141  1
```

```
;  765          1142  1 GLOBAL ROUTINE RESET_UNIT : COMMON_CALL NOVALUE =
;  766          1143  1
;  767          1144  1 !++
;  768          1145  1 !
;  769          1146  1 ! FUNCTIONAL DESCRIPTION:
;  770          1147  1 !
;  771          1148  1 !       This routine resets the unit so that after an error message
;  772          1149  1 !       the same unit is choosen for mount
;  773          1150  1 !
;  774          1151  1 !
;  775          1152  1 ! CALLING SEQUENCE:
;  776          1153  1 !
;  777          1154  1 ! INPUT PARAMETERS:
;  778          1155  1 !       NONE
;  779          1156  1 !
;  780          1157  1 ! IMPLICIT INPUTS:
;  781          1158  1 !       NONE
;  782          1159  1 !
;  783          1160  1 ! OUTPUT PARAMETERS:
;  784          1161  1 !       NONE
;  785          1162  1 !
;  786          1163  1 ! IMPLICIT OUTPUTS:
;  787          1164  1 !       NONE
;  788          1165  1 !
;  789          1166  1 ! ROUTINE VALUE:
;  790          1167  1 !       NONE
;  791          1168  1 !
;  792          1169  1 ! SIDE EFFECTS:
;  793          1170  1 !       NONE
;  794          1171  1 !
;  795          1172  1 !--
;  796          1173  1
;  797          1174  2     BEGIN
;  798          1175  2
;  799          1176  2     EXTERNAL REGISTER
;  800          1177  2         COMMON_REG;
;  801          1178  2
;  802          1179  2     IF .CURRENT_VCB[VCB$W_RVN] NEQ 0
;  803          1180  2     THEN
;  804          1181  2         CURRENT_VCB[VCB$W_RVN] = .CURRENT_VCB[VCB$W_RVN] - 1
;  805          1182  2     ELSE
;  806          1183  2         CURRENT_VCB[VCB$W_RVN] = .BBLOCK[.CURRENT_VCB[VCB$L_RVT], RVT$B_NVOLS]
;  807          1184  2         - 1;
;  808          1185  2
;  809          1186  1     END;


                                        0000 00000          .ENTRY  RESET_UNIT, Save nothing                    ; 1142
                                 0E  AB B5 00002            TSTW    14(CURRENT_VCB)                             ; 1179
                                     09 12 00005            BNEQ    1$                                          ;
                           50        20 AB D0 00007         MOVL    32(CURRENT_VCB), R0                         ; 1183
                        0E AB        08 A0 9B 0000B         MOVZBW  11(R0), 14(CURRENT_VCB)                     ; 1184
                                 0E  AB B7 00010 1$:        DECW    14(CURRENT_VCB)                             ;
                                     04 00013               RET                                                 ; 1186
```

```
; Routine Size:  20 bytes,    Routine Base:  $CODE$ + 04B5

;  810          1187  1
;  811          1188  1 END
;  812          1189  1
;  813          1190  0 ELUDOM




;                                  PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                         1225  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                               Library Statistics
;
;                                        -------- Symbols --------      Pages       Processing
;       File                             Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        82         0      1000          00:01.8




;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:NXTVOL/OBJ=OBJ$:NXTVOL MSRC$:NXTVOL/UPDATE=(ENH$:NXTVOL)

; Size:          1204 code + 21 data bytes
; Run Time:         00:24.7
; Elapsed Time:     00:53.4
; Lines/CPU Min:     2891
; Lexemes/CPU-Min: 19594
; Memory Used:  252 pages
; Compilation Complete
```

LOCKDB LIS

LOGIO LIS

MATCHNAME LIS

IODONE LIS

LOCKDN LIS

NXTVOL LIS

MAIL LIS

MOUVOL LIS

OPRCOM LIS

INIMTA LIS

MODIFY LIS

PARSE LIS